Functionless batch processing with EB

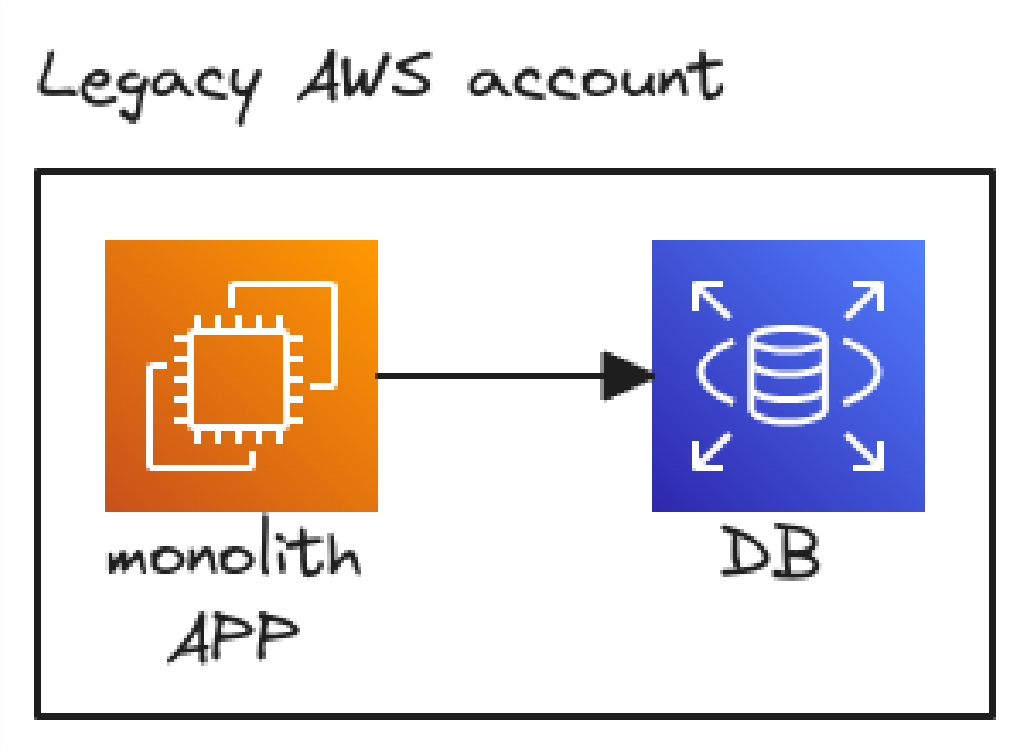
1. **About me**
2. **Situation**
3. **What we did**
4. **What we learnt**

About me

- I am Carlos Angulo
- I work at Ohpen as a Platform Engineer
- I have more than 8 years of experience
- You can find me on LinkedIn or GitHub

Situation

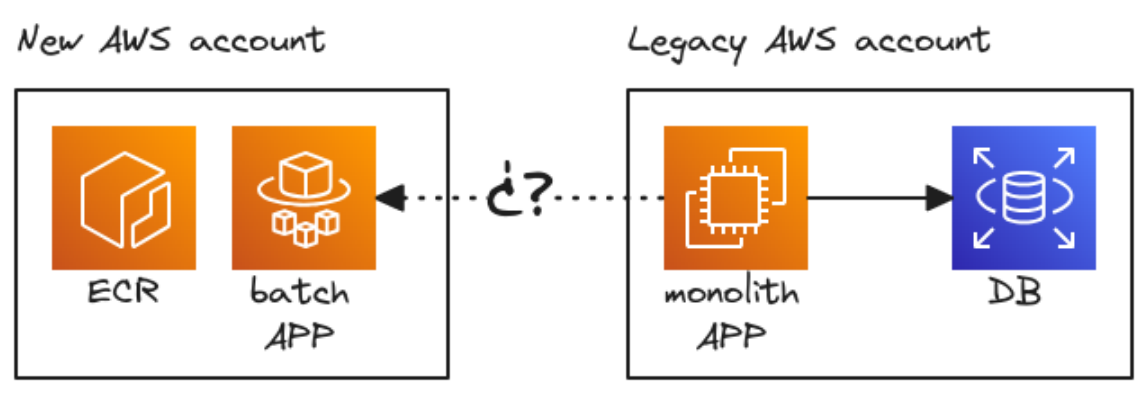
We have a legacy monolith app running in EC2 + RDS



Dockerization

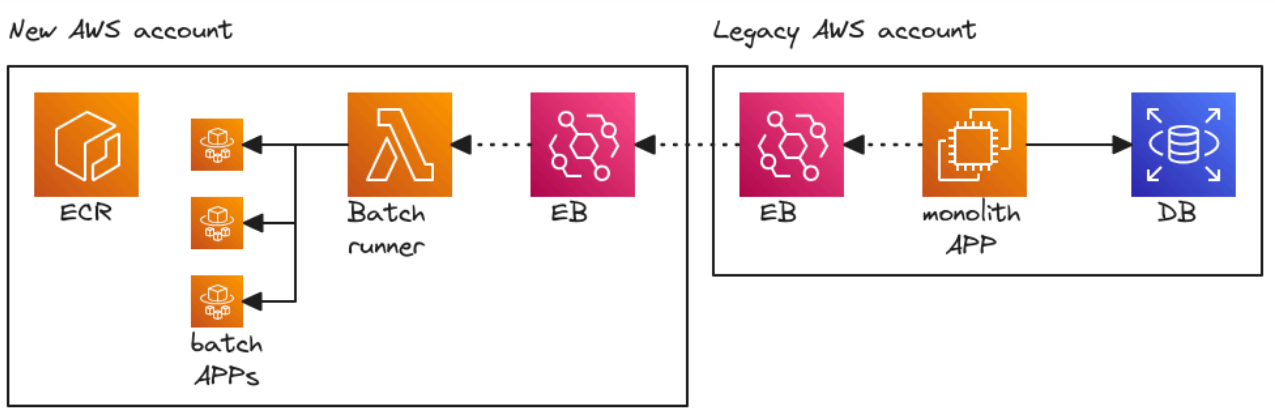
We dockerized some batch apps in a new account

How would you trigger those apps from the monolith? 🤔



What we did

We decide to trigger batches based on events



```
public async Task RunBatch(EventData detail, string traceId)
{
    var ecsConfig = await _configProvider.GetConfig(detail, traceId);
    var request = new RunTaskRequest
    {
        Cluster = ecsConfig.ClusterName,
        TaskDefinition = ecsConfig.TaskDefinition,
        NetworkConfiguration = new NetworkConfiguration
        {
            ... // omitted for brevity
        },
        Overrides = new TaskOverride
        {
            ContainerOverrides = new List<ContainerOverride>
            {
                new()
                {
                    Name = ecsConfig.TaskDefinition,
                    Command = ecsConfig.CommandParams,
                    Environment = ecsConfig.EnvironmentVariables
                        .Select(source => new EcsKeyValuePair() { Name = source.Key, Value = source.Value })
                        .ToList()
                }
            }
        }
    };

    var response = await _ecsClient.RunTaskAsync(request);
}
```

```
1  public async Task<EcsBatchConfig> GetConfig(EventData detail, string traceId)
2  {
3      var batchInfo = _supportedBatches.FirstOrDefault(b => b.BatchTypeId == detail.BatchTypeId);
4      if (batchInfo == null)
5          throw new NotImplementedException("Unsupported batch type");
6
7      var conf = _batchConfigurationProvider.Get(batchInfo.Id)
8      var config = new EcsBatchConfig(
9          conf.ClusterName,
10         conf.TaskDefinition,
11         ... // omitted for brevity
12         new List<string> { CommandName, CommandParamBatchTypeId, detail.BatchTypeId.ToString() }
13     );
14     config.EnvironmentVariables.Add(EnvVarTraceId, traceId); // For monitoring batch
15     if (detail.Parameters?.Any() ?? false)
16         // expose event parameters to batch
17         config.EnvironmentVariables.Add(EnvVarBatchParameters, detail.Parameters);
18     return config;
19 }
```

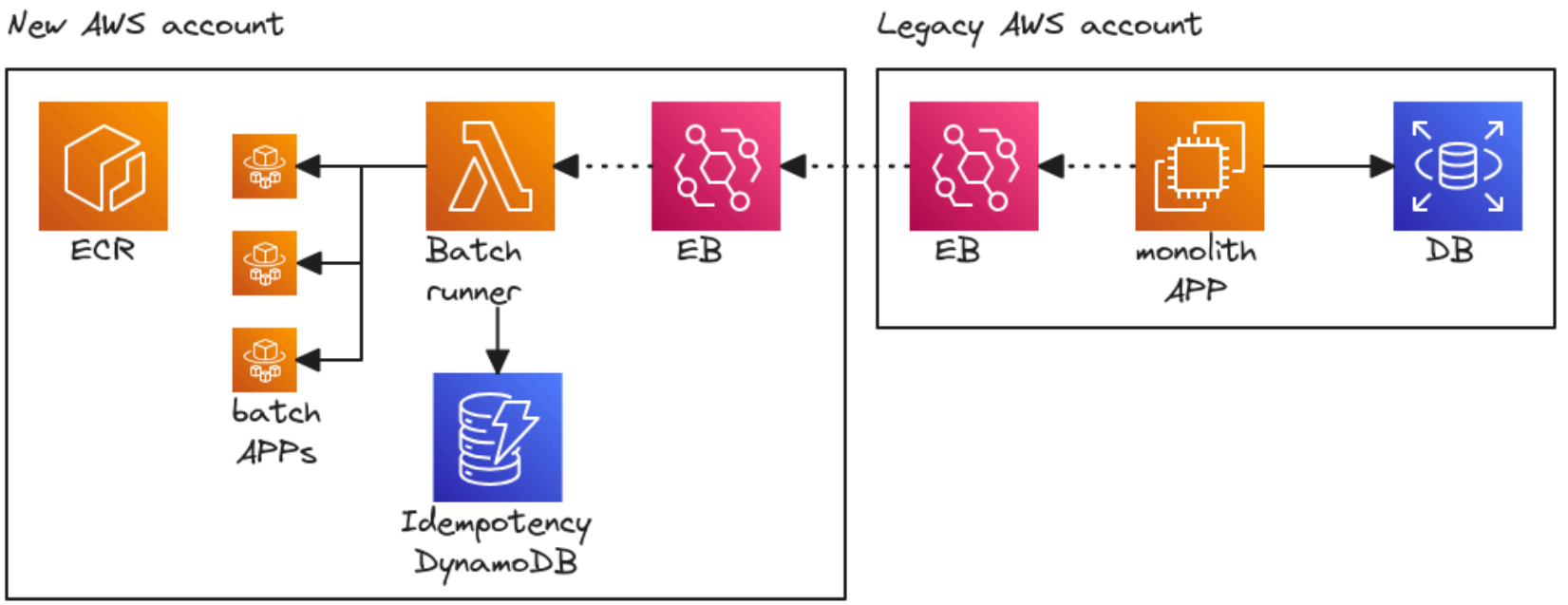
```
execute_ecs_tasks = {
  effect = "Allow"
  actions = [
    "ecs:RunTask",
    "iam:PassRole"
  ]
  resources  = ["batch-apps-*"]
  principals = []
}

```

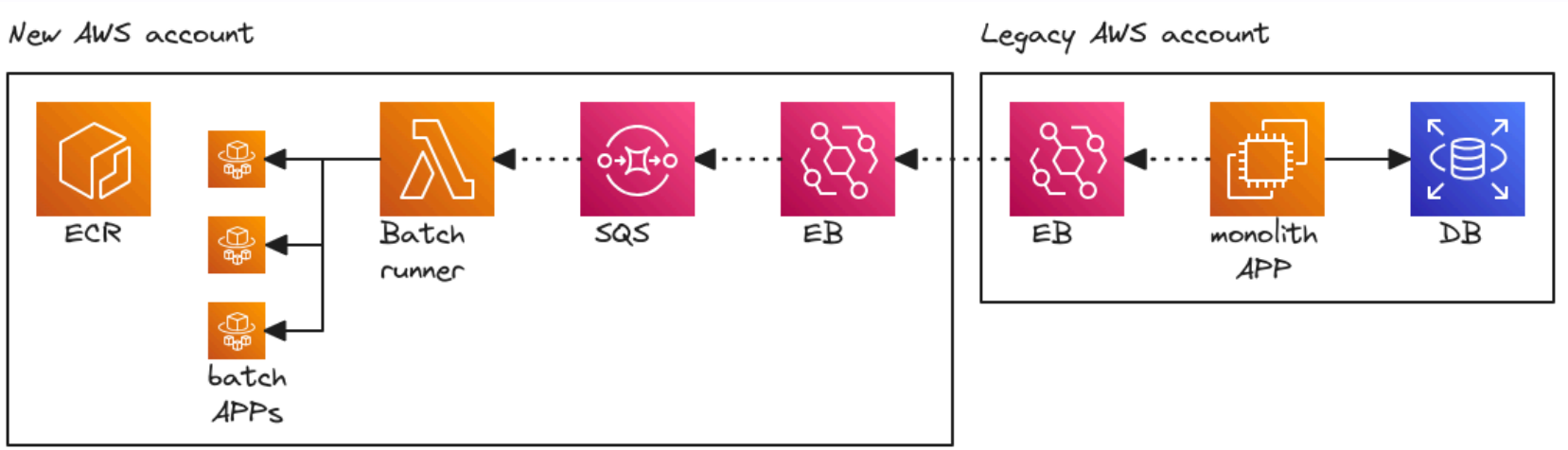
wait

What about duplicated events?

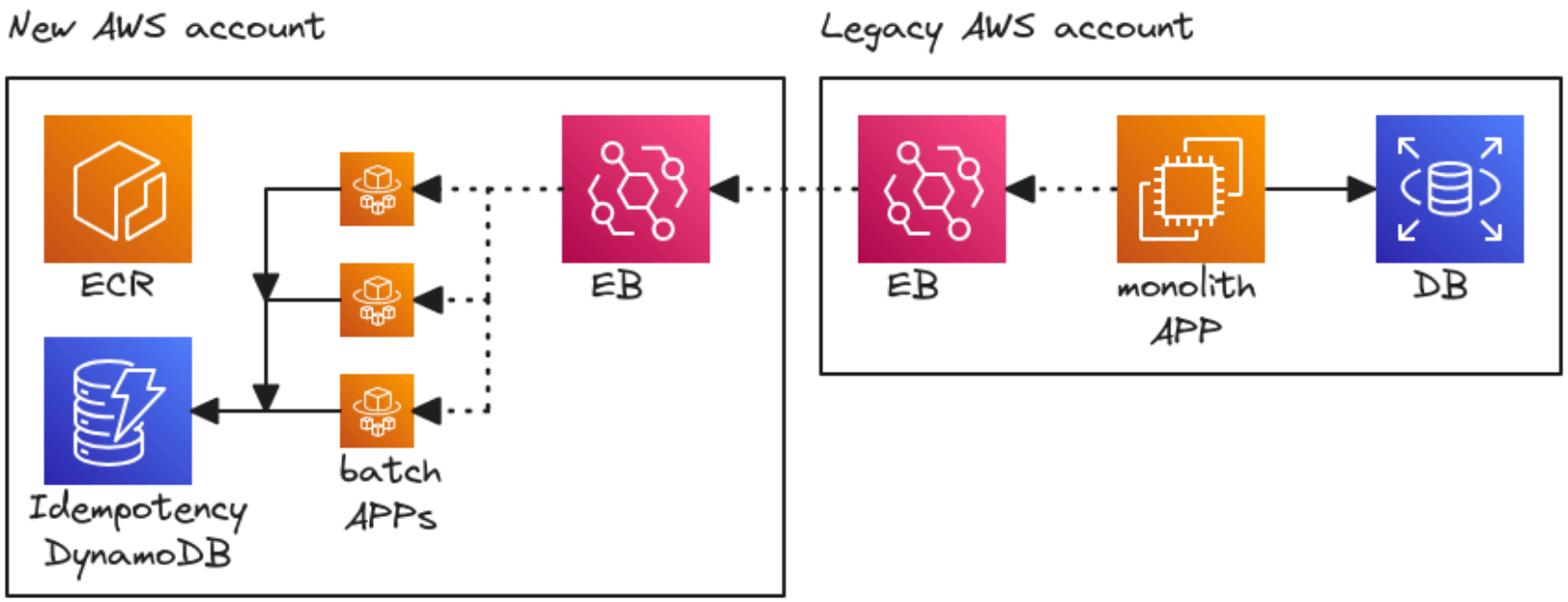
Idempotency with DynamoDB



Idempotency with SQS



simpler solution?



What we learn

- lambda (+ SQS) offers more control
- EB + ECS is simpler but more expensive and adds logic to batch apps

Questions?